**Введение**

На учебную практику по разработке и сопровождению программного обеспечения была поставлена задача «Разработка программы на 1С для ведения учёта о товаре на складах, продажах и контрагентах».

Цель учебной практики заключается в разработке программы, которае облегчит организацию работы небольшого предприятия по продаже каких-либо товаров.

Создаваемая программа предназначена для работы самого сотрудника предприятия.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название «Анализ задачи». В нем Вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно- экономической сущности. Также в этом разделе Вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе.

В разделе «Проектирование задачи» будут рассмотрены основные аспекты разработки программного продукта. В подразделе «Инструменты разработки» будет рассмотрена среда, в которой создаётся данный проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет чётко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

«Реализация» – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации сайта. В этом разделе будут чётко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвёртый раздел – «Тестирование». В нем будет описано полное и функциональное тестирование данного проекта, т.е. будет протестирован каждый пункт меню, каждая операция, которая выполняется приложением. «Заключение» будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В «Списке используемых источников» будет приведён список используемых при разработке источников. В «Приложение А» к пояснительной записке приведены UI и UX модели пользовательского интерфейса.

1. **Анализ задачи**
   1. **Постановка задачи** 
      1. **Организационно-экономическая сущность задачи**

**Наименование задачи**: Разработка программы на 1С для ведения учёта о товаре на складах, продажах и контрагентах.

**Цель разработки:** создание создание программы, которая позволит улучшить систему учёта товаров на складах, учёта продаж.

**Назначение:** программный продукт разрабатывается на сотрудников небольших компаний в сфере продаж чего-либо

**Предметная область:** улучшение работы небольших организаций в сфере продаж стройматериалов.

**Периодичность использования:** зависит от нужд потребителя, может

использоваться ежедневно.

**Источники и способы получения данных**: записи в документах, справочниках, отчёты.

* + 1. **Функциональные требования**

Разрабатываемая программа позволит выполнить следующие действия:

* добавление новых записей в документы;
* изменение уже существующих записей в документах;
* добавление новой информации в справочниках;
* изменение уже существующей информации в справочниках;
* возможность посмотреть отчёт о расходах.
  + 1. **Описание процессов с входной, выходной и условно-постоянной информацией**

Вся информация, которой оперирует пользователь в процессе решения задачи подразделяется на :

– входную информацию;

– выходную информацию.

Входной информацией выступают данные в документах, выходные данные – отчёт.

* + 1. **Эксплуатационные требования**

**Требования к применению:** помогает быстро и удобно отслеживать работу предприятия.

**Требования к реализации:** Для реализации программы должен использоваться язык разработки 1С:Предприятие.

**Требования к надежности:** Система может быть недоступна не более чем 24 часа в год.

**Требования к интерфейсу:** При разработке программы будут использованы встроенные цвета интерфейса.

* 1. **Инструменты разработки**

Для разработки данной программы был использован язык программирования 1С: Предприятие.

Иные инструменты, используемые при разработке и написании сопутствующей документации:

* WEB-ресурс DRAW.IO – будет использоваться для создания графической части и разработки UML-диаграмм;
* Microsoft Office Word – для написания документации к программному продукту;
* WEB-ресурс Figma – будет использоваться для разработки UI\UX моделей;

Разработка проекта будет происходить на компьютере со следующими параметрами:

* Intel(R) Core(TM) i3-11800H @ 2.30GHz 2.30 GHz;
* объем ОЗУ 16Гб;
* объем места на жестком диске 512 GB;
* графический адаптер nvidia gforce 3050 ti;
* ОС Windows 10 Домашняя.

1. [**Проектирование задачи**](#_Достоинства_и_недостатки)
   1. **Выбор стратегии разработки и модели жизненного цикла**

Для разработки веб-ресурса следует выбрать стратегию разработки и модель жизненного цикла. Осуществляется выбор посредством составления таблиц:

Таблица 1 **-** Выбор модели жизненного цикла на основе характеристик требований.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Критерии категории требований** | **Каскадная** | **V-образная** | **RAD** | **Инкремент- ная** | **Быстрого прототипи- рования** | **Эволюцион- ная** |
| 1 Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| 2 Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3 Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| 4 Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| 5 Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| 6 Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7 Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |
| Итого | 2 | 2 | 5 | 3 | 5 | 5 |

**Вычисления:** 2 за каскадную, 2 за V- образную, 5 за RAD, 3 за инкрементную, 5 за быстрого прототипирования и 5 за эволюционную.

**Итог:** На основе результатов заполнения таблицы 1 подходящей является RAD модель и инкрементная модель.

Таблица 2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Критерии категории команды разработчиков проекта** | **Каскадная** | **V-образная** | **RAD** | **Инкремент- ная** | **Быстрого прототипи- рования** | **Эволюцион- ная** |
| 1 Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2 Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| 3 Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4 Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| 5 Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6 Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |
| Итого | 4 | 4 | 1 | 4 | 3 | 4 |

**Вычисления:** 4 за каскадную, 4 за V-образную, 1 за RAD, 4 за инкрементную, 3 за быстрого прототипирования и 4 за эволюционную.

**Итог:** На основе результатов заполнения таблицы 2 подходящими являются каскадная, V-образная и инкрементная модели.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей. | | | | | | |
| **Критерии категории коллектива пользователей** | **Каскадная** | **V-образная** | **RAD** | **Инкрементная** | **Быстрого прототипированияи- рования** | **Эволюционная** |
| 1 Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| 2 Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе  разработки? | Нет | Нет | Нет | Да | Да | Да |
| 3 Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| 4 Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |
| Итого | 1 | 1 | 1 | 2 | 3 | 3 |

**Вычисления:** 1 за каскадную, 1 за V-образную, 1 за RAD, 2 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

**Итог:** На основе результатов заполнения таблицы 3 подходящей является модель быстрого проектирования.

Таблица 4 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Критерии категории типов проекта и рисков** | **Каскадная** | **V-образная** | **RAD** | **Инкремент- ная** | **Быстрого прототипи- рования** | **Эволюцион- ная** |
| 1 Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| Продолжение таблицы 4 |  |  |  |  |  |  |
| 2 Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| 3 Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| 4 Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| 5 Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| 6 Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7 Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| 8 Является ли гра0фик сжатым? | Нет | Нет | Да | Да | Да | Да |
| 9 Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| 10 Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |
| Итого | 2 | 3 | 3 | 9 | 7 | 9 |

**Вычисления:** 2 за каскадную, 3 за V-образную, 3 за RAD, 9 за инкрементную, 7 за быстрого прототипирования и 9 за эволюционную.

**Итог:** На основе результатов заполнения таблицы 4 подходящей является инкрементная и эволюционная модели.

Таблица 5 – Подведение итогов со всех предыдущих таблиц.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ таблицы** | **Каскадная** | **V-образная** | **RAD** | **Инкремент- ная** | **Быстрогопрототипирования** | **Эволюционная** |
| 1 | 2 | 2 | 5 | 3 | 5 | 5 |
| 2 | 4 | 4 | 1 | 4 | 3 | 4 |
| 3 | 1 | 1 | 1 | 2 | 3 | 3 |
| 4 | 2 | 3 | 3 | 9 | 7 | 9 |
| Итого | 9 | 10 | 10 | 18 | 18 | 21 |

**Общий итог:** в итоге заполнения таблиц 1 – 4 наиболее подходящей является эволюционная модель.

* 1. **Разработка структуры сайта, системы меню, навигации**

Проектирование программы важно, так как это определяет пользовательский опыт и влияет на эффективность самой программы. Хорошо спроектированная программу обеспечивает удобство использования (рисунок 1).

Рисунок 1 – Cтруктура программы

* 1. **Разработка UML-диаграмм**
     1. **Диаграмма вариантов использования**

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. (рисунок 2).

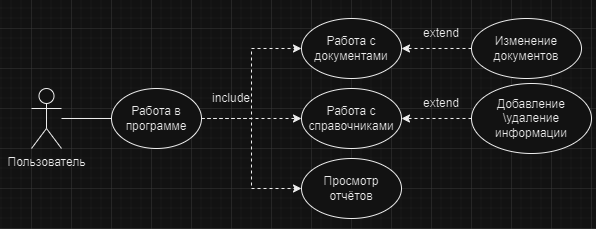


Рисунок 2 – Диаграмма вариантов использования

* + 1. **Диаграмма последовательности**

Диаграмма последовательности (англ. sequence diagram) — UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некойсущности) (рисунок 3).

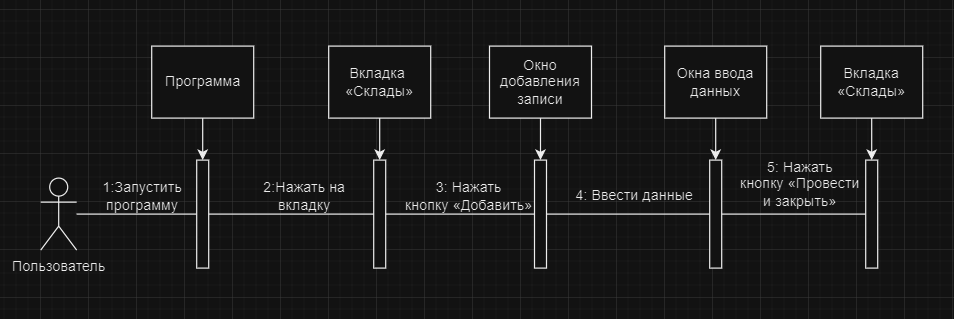


Рисунок 3 – Диаграмма последовательности добавления склада в справочник

* + 1. **Диаграмма деятельности**

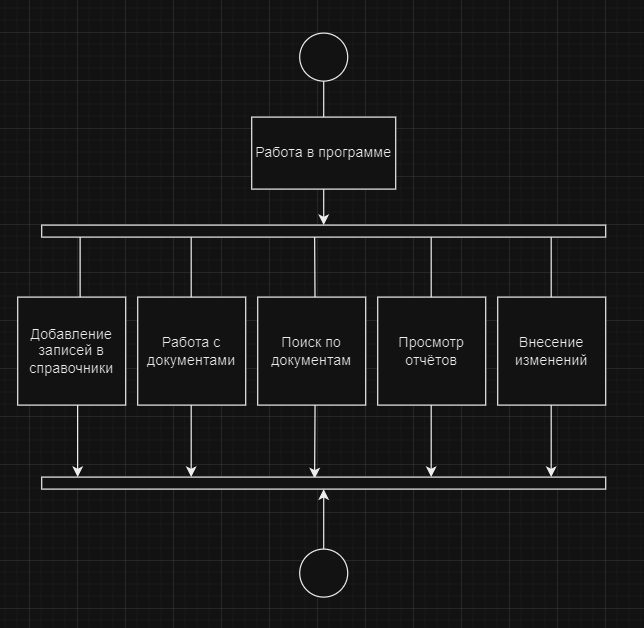
Диаграмма деятельности — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний (рисунок 4).

Рисунок 4 – Диаграмма деятельности для пользователя

* 1. **Разработка пользовательского интерфейса**

Важным элементом проектирования данного программного продукта является описание внешнего интерфейса разрабатываемой программы.

Для разработки визуального дизайн использовались сдержанные, мягкие цвета для удобства использования программного продукта.

Для организации эффективной работы пользователя нужно создать целостную программу данной предметной области, в которой все компоненты будут сгруппированы по функциональному назначению. При этом необходимо обеспечить удобный графический интерфейс пользователя. Программа должна позволить пользователю решать задачи, затрачивая значительно меньше усилий, чем при работе с разрозненными объектами. Все исходные данные будут разделены на несколько групп.

Прототип – это наглядная модель пользовательского интерфейса. В сущности, это «черновик» созданный на основе представления разработчика о потребностях пользователя. Итоговое отображение программы может отличаться от прототипа. C прототипами UX/UI можно ознакомиться в приложении А.

1. **Реализация**

Данная программа разработана на языке 1С: Предприятие. Данная программа хорошо подходит для реализации подобных задач как эта.

Далее будет информация о компонентах которые были использованы при реализации программы:

* Справочники – представляют собой структурированные списки данных, используемые для хранения информации. Они могут использоваться для хранения справочной информации, такой как клиенты, поставщики, товары и т. д. В 1C: Предприятие можно создавать справочники, определять их структуру и свойства, а также использовать для взаимодействия с другими объектами системы (Сперва создаём сами справочники и настраиваем их параметры. Далее создаём новую запись. После нажатия кнопки Провести и Закрыть сохраняется запись в справочнике);
* Документы – представляют собой основные объекты для работы с оперативными данными. Документы используются для отражения различных бизнес-событий и процессов в организации. Например, это могут быть договоры, счета, заказы, накладные и другие виды операций (Создаём документ и настраеваем его параметры. Далее отмечаем какое-то действие и нажимаем Провести и Закрыть);
* Отчёты - используются для анализа и представления информации, хранящейся в системе. Отчеты могут включать в себя различные графики, таблицы, группировки и фильтры, что позволяет пользователям получать нужную информацию в удобном виде (Создаём отчёт и настраеваем его параметры. Далее выбираем что хотим отобразить в отчёте и сохраняем выбор. Также можно настроить отображение отчёта, графиком, таблице или же диаграммой).

1. [**Тестирование**](#_Достоинства_и_недостатки)
   1. **Тесты на использование**

При разработке сайта «Личный кабинет психолога» многие возникающие ошибки и недоработки были исправлены на этапе реализации программного продукта. После завершения испытания реализации интернет-ресурса было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программного продукта в автономном режиме.

Таблица 8 – Тесты на использование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название теста** | **Действие** | **Ожидаемый результат** | **Физический результат** | **Результат тестирования** |
| Корректный запуск программы | Запустить ярлык программы | Запуск программы без ошибок | Запуск программы без ошибок | Выполнено |
| Кнопки вкладок | Нажатие на кнопку вкладки | Переход на соответствующую вкладку | Переход на соответствующую вкладку | Выполнено |
| Кнопки Провести и Закрыть | Нажатие на кнопку Провести и Закрыть | Успешное сохранение записи в компоненте | Успешное сохранение записи в компоненте | Выполнено |
| Кнопка Отчёты | Нажатие на кнопку Отчёты | Появление окна с списком отчётов | Появление окна с списком отчётов | Выполнено |

* 1. **Отчет о результатах тестирования**

В результате проведения тестирования выяснилось, что все ранее оговоренные функции и требования, были разработаны, а также протестированы. Тесты показали, что все функции работают правильно, следовательно, разработанную программу можно передать заказчику.

В ходе тестирования программного обеспечения программы на устройстве не было выявлено каких-либо ошибок.

1. **Руководство пользователя**

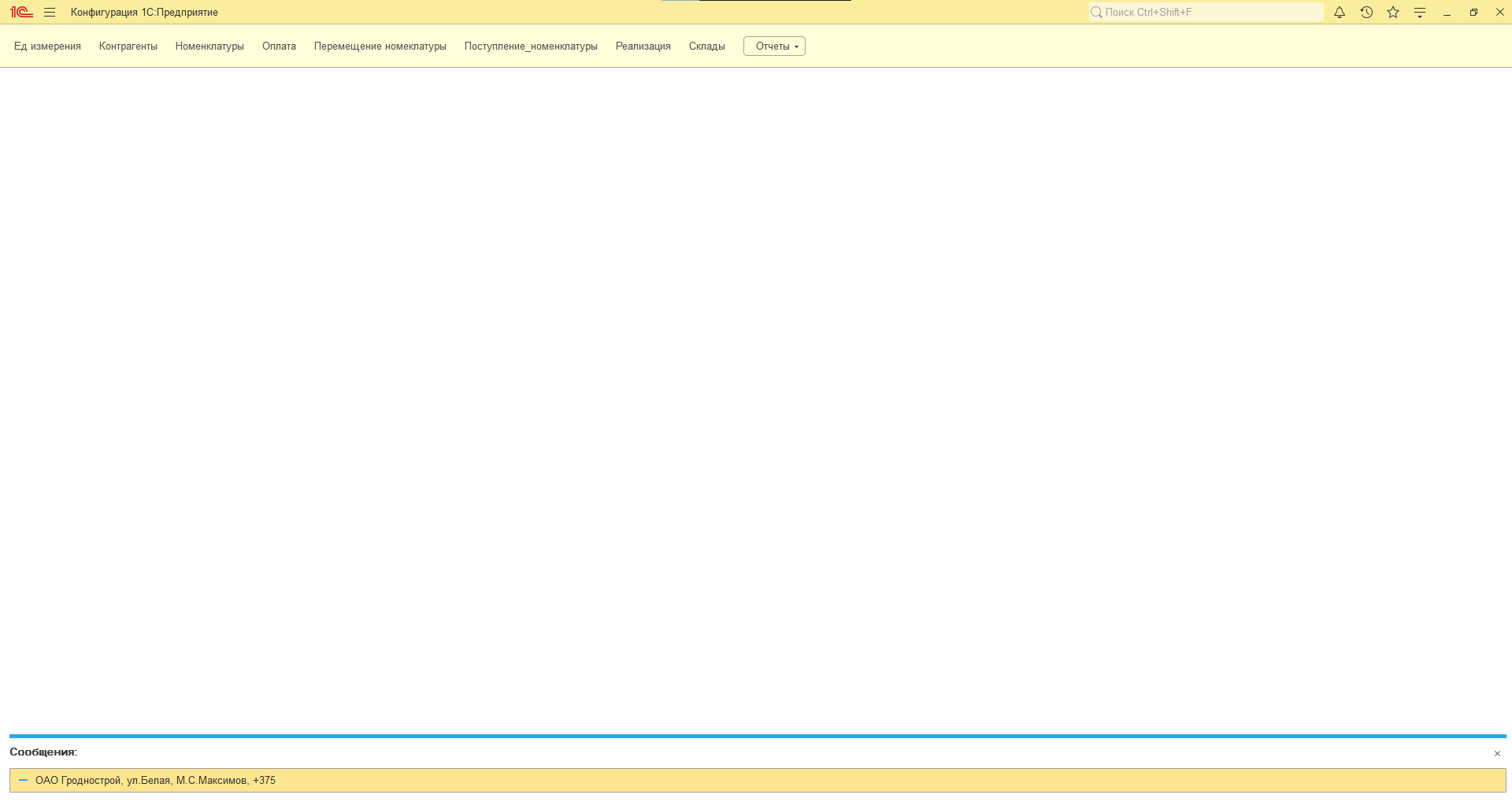
При входе в программу нас встречает главная страница с небольшой информацией и навигацией. Простой и удобный интерфейс (рисунок 5). Сверху на панели находится навигация по программе.

Рисунок 5 – Главная страница

На вкладках «Контрагенты», «Номенклатуры», «Склады», «Ед измерения» написана некоторая информация, которой можно пользоваться при работе с документами. Также в этих вкладках можно добавлять или удалять элементы (рисунок 6, 7, 8).

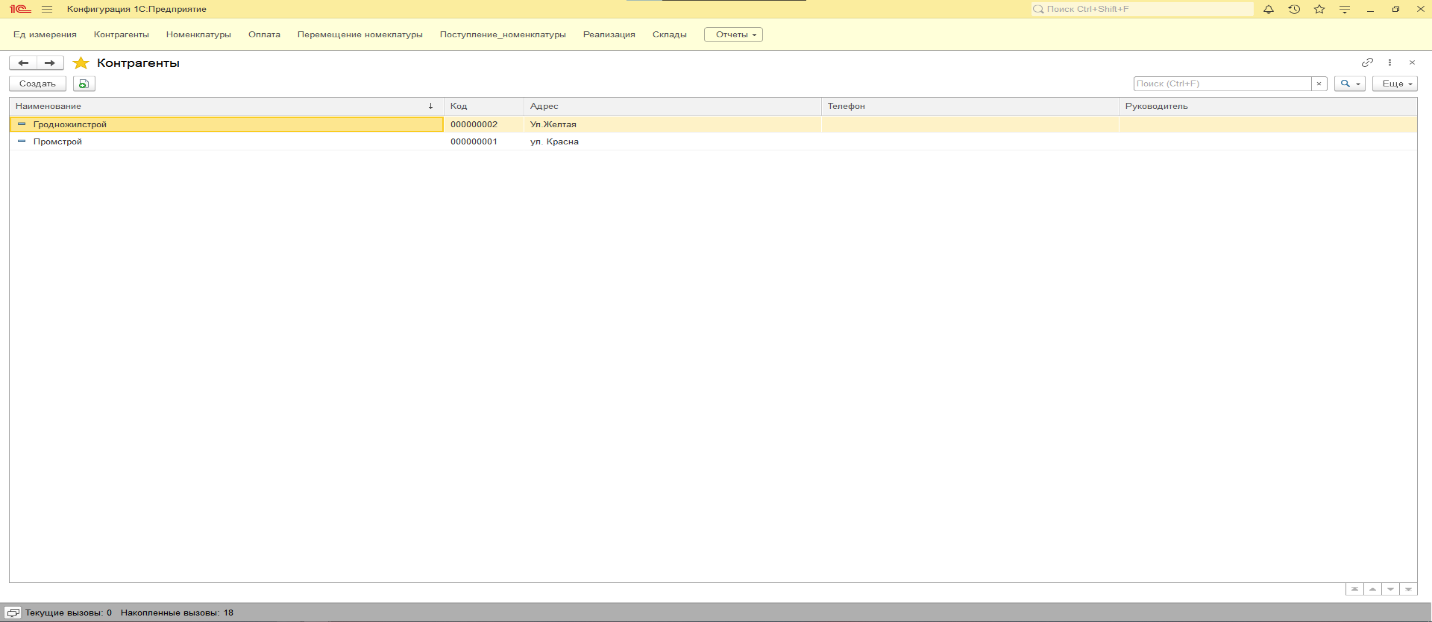
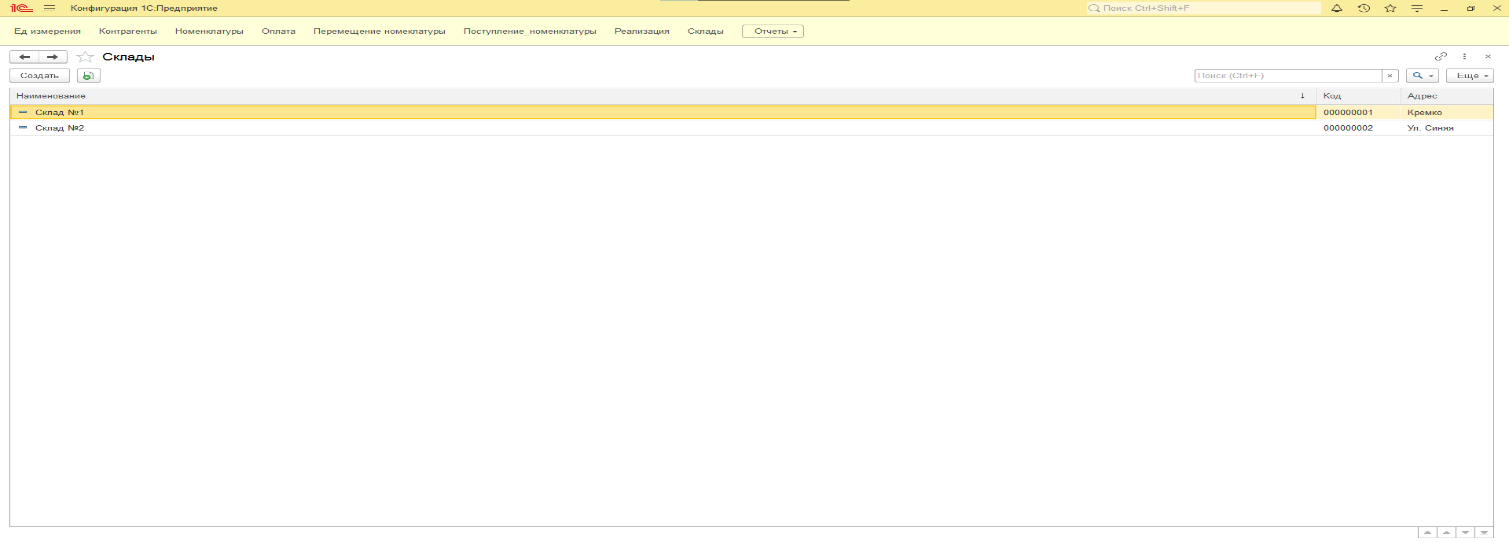


Рисунок 6 – Вкладка «Контрагенты»



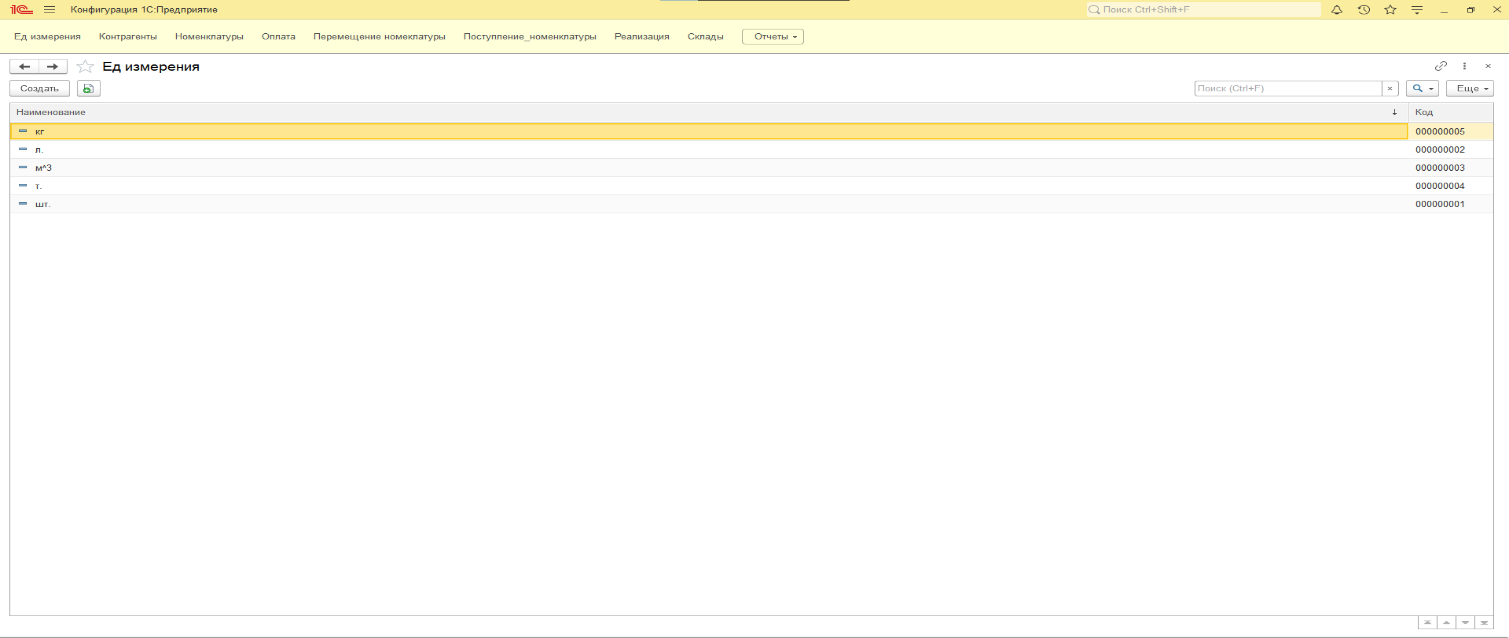
 Рисунок 7 – Вкладка «Склады»

Рисунок 8 – Вкладка «Ед измерения»

В остальных вкладках идёт основная работа, то есть работа в документах. В них, к примеру, можно отметить перемещение номенклатуры по складам. Также можно отметить перевод денег котрагентам в вкладке «Оплата». Также в программе можно просмотреть отчёты.

[**Заключение**](#_Toc406878088)

Целью данного проекта «Разработка программы на 1С для ведения учёта о товаре на складах, продажах и контрагентах» является разработка программы, которая позволит улучшать качество работы небольших предприятий, повышение эффективности работы сотрудников. Программа разрабатывается на пользователей, работающих на предприятиях, с целью упрощения их рабочей деятельности.

В целом при реализациипрограммы, были выполнены все условия, поставленные на начальном этапе разработки. Из чего мы можем сделать вывод, что программу можно передавать заказчику и вводить в эксплуатацию.

Верю, что данная программа поможет работникам небольших предприятий в сфере продаж.

**Список использованных источников**

1. Справочник 1C: Предприятия – Права на чтение
2. Сайта FIGMA для создания макетов - <https://www.figma.com/> Права на изменение\добавление – 15.12.2023

**Приложение А**

UX

